Assignment #3

Eric Rouse; Individual Programming 53

# Understanding the Problem

Write a program that calculates the integral of f(x) = 2\*x^2 numerically using two different methods, area by rectangles and area by trapezoids. Make sure to use full error checking and handling on all the places that a user inputs data. Let the user re-enter data as needed.

# Devising a Plan/Design

Here is the pseudocode that I came up with.

1. Prompt the user for the starting and ending points of the integration area.  
   {reprompt if not an integer or InputMismatchDetected}
2. Prompt the user to choose a method of numeric integration, either   
   [1] Rectanglular, or  
   [2] Trapezoidal, or  
   [3] Both  
   {reprompt if one of these is not selected or InputMismatchDetected}
3. Let user choose the amount of rectangles and trapezoids used. These are independently determined in the case that both are chosen.   
   {reprompt if not an integer or InputMismatchDetected}
4. Print function being evaluated: **f(x)=2x2**
5. Print starting and ending points
6. Print number of rects/traps used
7. Print the calculated areas by the selected methods.
   1. Area by rectangles:
      1. Base = (end point – start point)/# rects
      2. Height = f(sum of previous bases)
      3. Total Area =
   2. Area by trapezoids
      1. Base = (end point – start point)/# traps
      2. Height1 = f(sum of previous bases)
      3. Height2 = f(sum of previous bases + current base)

Total Area =

# Looking Back/Self-Reflection

Wow. Error checking is a real pain. At least 80% of my program has to do with making sure the data that I let the user input is correct and won’t cause a crash. I mean, really, the actual integral calculation is two lines of code. The rectangular method for instance:

**for** (**float** i = a; i <= b;i+=rsd){

rectarea += rsd\*2.0\*Math.*pow*(i, 2.0);

}

Everything else is about get in data, making sure it is good and prompting the user along!

To check the output I worked through every possible path in the code and entered faulty data where I could. To check the integration I ran several different possibilities and verified them by hand using the relation:

evaluated over the interval.

I learned how to apply error checking to what seems like trivial inputs and how hard that can be.